Accelerating Design Optimization with Model Order Reduction

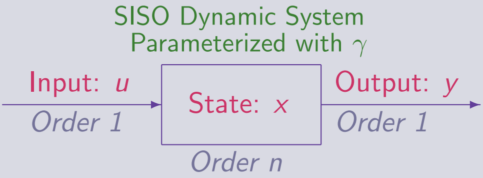
Presenter: Yao Yue

Promotor: Karl Meerbergen

July 9, 2010

Outline

Parameterized Vibration Systems



$$PDE \xrightarrow{\text{Discretization}} \begin{cases} \big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)x(\omega,\gamma) = f\, u(\omega,\gamma), \\ y(\omega,\gamma) = \ell^* x(\omega,\gamma), \end{cases}$$

$\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ denotes k design parameters. Assume $u \equiv 1$.

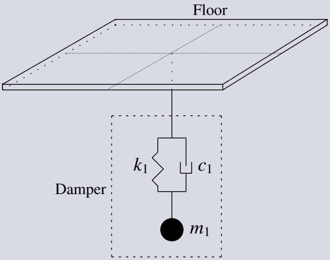
$K(\gamma)$: stiffness; $C(\gamma)$: damping; $M(\gamma)$: mass.

Discretization $\Longrightarrow$ The system order n is usually large.

- **Model Order Reduction**: Reduce the order of x.
- **Optimal Design**: Choose γ to best meet our design objective.

Example: Floor Damper Design

Problem Statement



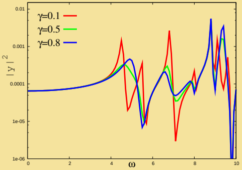
- **Goal**: Use a damper to decrease the vibration of a floor.
- **Floor**: Described with a shell element finite element model.
- **Damper**: Described with a K-C-M model.
- **Design Objective:** Minimize the vibration at a point on the floor by choosing k_1 and c_1.

Mathematical Formulation

$$\begin{cases} \left(K_0 + (k_1 + i\omega c_1)K_1 - \omega^2 M\right) x = f, \\ \hfill y = \ell^* x. \end{cases}$$

Optimal Design of Vibration Systems

Optimal Design: Optimize the FRF



Each γ value corresponds with an FRF.

- Minimize the highest peak of the FRF.
- Minimize the overall energy of the FRF.

Two Phases in Design Optimization

- **Outer Phase:** $\min_{\gamma} g(\gamma)$.
- **Inner Phase:** $$g(\gamma) = \begin{cases} \max\limits_{\omega_L \le \omega \le \omega_H} |y(\omega,\gamma)|^2, & (\infty\text{-norm optimization}), \\ \int_{\omega_L}^{\omega_H} |y(\omega,\gamma)|^2 \, \mathrm{d}\omega, & (2\text{-norm optimization}). \end{cases}$$

Algorithm: Inner Phase

- ∞-norm Case: Grid Search + Quasi-Newton;
- 2-norm Case: Trapezoidal rule.

Algorithm: Outer Phase

Quasi-Newton type method.
Note: Non-smooth in ∞-norm case.

More on Quasi-Newton

All Quasi-Newton iterations use the backtracking strategy with an Armijo condition to force convergence.

- Smooth Function: Super-linear convergence rate.
- Non-smooth Function: The convergence rate may degrade. Needs more backtracking steps.

Computational Analysis

Let $\mathcal{L}(\omega,\gamma) = K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)$.

$$y = \ell^* \mathcal{L}(\omega,\gamma)^{-1} f, \quad \frac{\partial y}{\partial \omega} = \ell^* \mathcal{L}(\omega,\gamma)^{-1} \big(2\omega M(\gamma) - iC(\gamma)\big) \mathcal{L}(\omega,\gamma)^{-1} f$$

$$\frac{\partial y}{\partial \gamma_j} = \ell^* \mathcal{L}(\omega,\gamma)^{-1} \left(-\frac{\partial K(\gamma)}{\partial \gamma_j} - i\omega \frac{\partial C(\gamma)}{\partial \gamma_j} + \omega^2 \frac{\partial M(\gamma)}{\partial \gamma_j} \right) \mathcal{L}(\omega,\gamma)^{-1} f$$

- Computing derivatives is cheap if we have already computed the function values.
  - Quasi-Newton type methods are suitable.
- Expensive: LU factorization of $\mathcal{L}(\omega,\gamma)$ for many (ω,γ) values.
  - Our Solution: Use Model Order Reduction (MOR) in optimization.

Outline

Basic Idea

$$\boxed{\textit{High Order System}} \xrightarrow[\textit{Krylov subspace}]{\textit{Project to a}} \boxed{\textit{Low Order System}}$$

Order n $\quad n \gg k \quad$ Order k

Krylov subspace: $\mathcal{K}_k\{A, b\} \triangleq \operatorname{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}$.

A numerically stable method to generate it: Arnoldi Process.

First Order Linear system with no design parameters

$$\begin{cases} (K - \alpha M)x = f \\ y = \ell^* x \end{cases}, \quad y = \ell^*(K - \alpha M)^{-1} f \triangleq \sum_{i=0}^{\infty} m_i \alpha^i$$

m_i is called the i-th moment of y.

$$m_i = \ell^* \underbrace{\left[(K^{-1}M)^i K^{-1} b\right]}_{\triangleq\, r_i,\ \text{the } i\text{-th moment of } x,} = \underbrace{\left[(K^{-*}M^*)^i K^{-*} \ell\right]}_{\triangleq\, \ell_i,\ \text{the } i\text{-th moment of } t,}{}^* b, \qquad \left(t = (K - \alpha M)^{-*} \ell\right).$$

Idea:

- Approximate x in the k-th right Krylov subspace $\mathcal{K}_k(K^{-1}M, K^{-1}b)$;
- Approximate t in the k-th left Krylov subspace $\mathcal{K}_k(K^{-*}M^*, K^{-*}\ell^*)$;
- Or both?

Projection Based MOR

Define $W_k, V_k \in \mathbb{C}^{n\times k}$, and $\hat{K} = W^*KV$, $\hat{M} = W^*MV$, $\hat{f} = W^*f$, $\hat{\ell} = V^*\ell$.
We can assemble the following reduced order system:

$$\begin{cases} (\hat{K} - \alpha \hat{M})z = \hat{f} \\ \hat{y} = \hat{\ell}^* z \end{cases}$$

Question: How to choose W_k and V_k to obtain good accuracy?

Krylov method: Moment Matching Property

Theorem:

1. If colspan$\{V_k\} \supseteq \mathcal{K}_k(K^{-1}M, K^{-1}b)$, the first k moments of y and $\hat{y}$ match;
2. If colspan$\{W_k\} \supseteq \mathcal{K}_k(K^{-*}M^*, K^{-*}\ell^*)$, the first k moments of y and $\hat{y}$ match;
3. If colspan$\{V_k\} \supseteq \mathcal{K}_k(K^{-1}M, K^{-1}b)$ and colspan$\{W_k\} \supseteq \mathcal{K}_k(K^{-*}M^*, K^{-*}\ell^*)$, the first $2k$ moments of y and $\hat{y}$ match.

Krylov Method $\Rightarrow$ Padé type approximation.

Case 1 and 2 are called one-sided methods.
Case 3 is called a two-sided method.

MOR on Second Order Systems: SOAR

A Second Order System:

$$\begin{cases} (K + i\omega C - \omega^2 M)x = f, \\ y = \ell^* x. \end{cases}$$

Conventional MOR method: Arnoldi process on the linearized system.

Disadvantages:

- The system order is doubled;
- The second order structure is not preserved.

SOAR stands for Second Order ARnoldi Process [Bai & Su, 05].

$$\begin{cases} (K + i\omega C - \omega^2 M)x = f, \\ y = \ell^* x. \end{cases} \xrightarrow{SOAR} \begin{cases} (\hat{K} + i\omega \hat{C} - \omega^2 \hat{M})z = \hat{f}, \\ \hat{y} = \hat{\ell}^* z. \end{cases}$$

Order n System $\quad n \gg k \quad$ *Order k System*

- SOAR projects the system to a larger subspace $\Longrightarrow$ at least as accurate;
- Linearization is not needed and the second order structure is preserved.

Parameterized Model Order Reduction (PMOR)

A linear system parameterized with γ:

$$\begin{cases} \big(G_0 + \gamma G_1 + s(C_0 + \gamma C_1)\big)x = b \\ y = \ell^* x \end{cases}$$

Idea of Krylov type PMOR: Multiparameter moment matching.

Let $y = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} m_i^j s^i \gamma^j$,

m_i^j is called the (i, j)-th 2-parameter moment w.r.t y.

PIMTAP

PIMTAP: Parameterized Interconnect Macromodeling via a Two-directional Arnoldi Process [Li, Bai, Su & Zeng, 08].

$$\begin{cases} \big(G_0 + \gamma G_1 + s(C_0 + \gamma C_1)\big)x = b \\ y = \ell^* x \end{cases} \qquad \textit{Order } n \textit{ System}$$

$$\Downarrow \textit{PIMTAP} \qquad\qquad n \gg k$$

$$\begin{cases} (\hat{G}_0 + \gamma \hat{G}_1 + s(\hat{C}_0 + \gamma \hat{C}_1))\hat{x} = \hat{b} \\ \hat{y} = \hat{\ell}^* \hat{x} \end{cases} \qquad \textit{Order } k \textit{ System}$$

Moment Matching Pattern

PIMTAP generates the left and right 2-parameter Krylov subspaces according to Moment Matching Pattern.

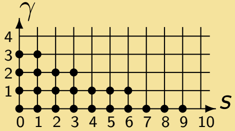


Figure: Example of Moment Matching Pattern.

All moments specified by the moment matching pattern will be matched for y and $\hat{y}$.

Advantages of PIMTAP

- Structure preserving;
- Numerically stable.

Use (P)MOR for Design Optimization

How?

Now we can:

- Reduce on the frequency with MOR;
- Reduce on both the frequency and the design parameters with PMOR.

When we have several design parameters, it is not realistic to reduce on all of them. Fix some of them.

- A free parameter is allowed to change in the reduced model.
- A fixed parameter is set to a specific value in the reduced model.

Importance about Computing Derivatives via (P)MOR

- Computing y, $\frac{\partial y}{\partial \omega}$ and $\frac{\partial y}{\partial \gamma_j}$ require the LU factorization of the same matrix, which is computationally dominant.
- MOR must be able to compute all of them to be efficient in reducing the computational cost.

Outline

Derivative Computation via (P)MOR

A General System

$$\begin{cases} A(p_1, p_2, \ldots, p_l)x = b, \\ y = \ell^* x. \end{cases}$$

Denote the $(i_1, i_2, \ldots, i_l)$-th moments of

- y: $m(i_1, i_2, \ldots, i_l)$;
- $x = A^{-1}(p_1, p_2, \ldots, p_l)b$: $r(i_1, i_2, \ldots, i_l)$;
- $t = A^{-*}(p_1, p_2, \ldots, p_l)\ell$: $\ell(i_1, i_2, \ldots, i_l)$.

Left and Right Krylov Subspace

- The $(i_1, i_2, \ldots, i_l)$-th right Krylov subspace:

  $\mathcal{K}^{(r)}(i_1, i_2, \ldots, i_l) = \text{span}\,\{r(j_1, j_2, \ldots, j_l) \,|\, j_1 \leq i_1, j_2 \leq i_2, \ldots, j_l \leq i_l\}$;
- The $(i_1, i_2, \ldots, i_l)$-th left Krylov subspace:

  $\mathcal{K}^{(l)}(i_1, i_2, \ldots, i_l) = \text{span}\,\{\ell(j_1, j_2, \ldots, j_l) \,|\, j_1 \leq i_1, j_2 \leq i_2, \ldots, j_l \leq i_l\}$.

Moment Matching Properties w.r.t Free Parameters

Differentiate $y(p_1, p_2, \ldots, p_l)$ on p_i once

$\Longrightarrow$ Shift the moment matching pattern back 1 step in p_i direction.

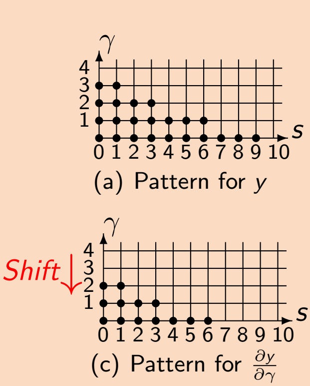


Figure: Derivative Computation of a System Containing Two Parameters.

(P)MOR w.r.t Fixed Parameters

Consider an additional fixed parameter q.

$$\begin{cases} A(p_1, p_2, \ldots, p_l | q)x = b, \\ y = \ell^* x. \end{cases} \xrightarrow{\text{MOR with } q=q_0} \begin{cases} \hat{A}(p_1, p_2, \ldots, p_l | q = q_0)\hat{x} = \hat{b}, \\ \hat{y} = \hat{\ell}^* \hat{x}. \end{cases}$$

Theorem

If we reduce the system with

- Right Krylov subspace $\mathcal{K}^{(r)}(i_1, i_2, \ldots, i_l)$.
- Left Krylov subspace $\mathcal{K}^{(l)}(i_1, i_2, \ldots, i_l)$. (Same indices).

then, moments with order up to $(i_1, i_2, \ldots, i_l)$ are matched for $\left.\frac{\partial y}{\partial q}\right|_{q=q_0}$ and $\left.\frac{\partial \hat{y}}{\partial q}\right|_{q=q_0}$.

Outline

Integrate (P)MOR into Optimization

High Computational Cost of the Inner Phase

∞-norm Case

- Computing y for all grid points in the coarse grid search;
- Computing y and $\frac{\partial y}{\partial \omega}$ for each step in the one-dimensional Quasi-Newton refinement;
- Computing ∇g ($g(\gamma) = \max_{[\omega_L, \omega_H]} |y(\omega, \gamma)|^2$) at the optimizer found.

2-norm Case

- Computing y and $\frac{\partial y}{\partial \gamma_i}$ ($i = 1, 2, \ldots, l$) for all interpolation points in numerical integration.

Use (P)MOR to Reduce the Cost

MOR can compute all of them with moment matching properties

$\Longrightarrow$ Use (P)MOR to reduce the computational cost.

We propose two frameworks: MOR Framework and PMOR Framework.

MOR Framework

- In an Inner Phase, all design parameters are fixed, we can use SOAR to reduce it.
- MOR Framework uses one SOAR reduced model for each Inner Phase.

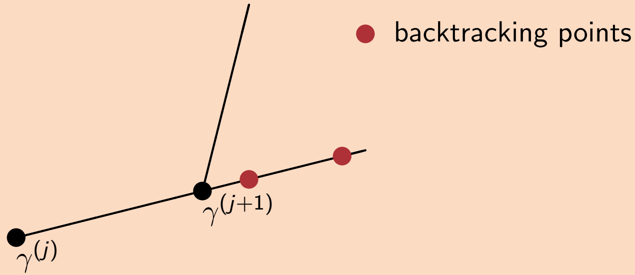


Figure: Line Search Optimization in the Parameter Space

One SOAR for each point.

PMOR Framework

- In MOR Framework, we need several reduced models for a line search.
- If we also reduce on the line search direction, we can compute them with only one reduced model.

Assume we are at $\gamma^{(j)}$ and the direction of the next step is $d^{(j)}$. On this direction, the system can be written as

$$\begin{cases} (K(\gamma^{(j)} + \alpha d^{(j)}) + i\omega C(\gamma^{(j)} + \alpha d^{(j)}) - \omega^2 M(\gamma^{(j)} + \alpha d^{(j)}))x = f, \\ y = \ell^* x. \end{cases}$$

If $K(\gamma)$, $C(\gamma)$ and $M(\gamma)$ are all matrix polynomials, it can be linearized as:

$$\begin{cases} \left(\tilde{K} + \alpha\tilde{C} + i\omega\tilde{M}\right)\tilde{x} = f, \\ y = \tilde{\ell}^* \tilde{x}, \end{cases}$$

Can be reduced with two-sided PIMTAP.

PMOR Framework

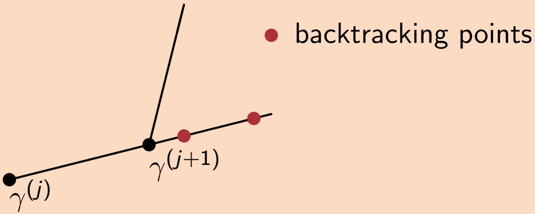


Figure: Line Search Optimization in the Parameter Space

One PIMTAP for all these point.

MOR Framework vs PMOR Framework

- When backtracking is frequent, PMOR Framework is more efficient; otherwise, MOR Framework is more efficient.
- For Quasi-Newton type methods, MOR Framework is more efficient for smooth objective functions, and PMOR Framework is efficient for non-smooth objective functions.

Outline

Floor Damper Design Problem: ∞-norm Optimization

Table: Low Order Example

	Direct method	MOR Framework	PMOR Framework
Matrix size	280	25	(15, 10, 8, 5)
Optimizer computed	(1.005e7, 9.853e4)	(1.005e7, 9.853e5)	(1.005e7, 9.853e4)
Optimized value	165.7502635	165.7502263	165.7501875
Number of Iterations	36	35	34
Backtracking Steps	149	126	110
CPU time	121s	23s	15s

Table: High Order Example

	Direct method	MOR Framework	PMOR Framework
Matrix size	29800	25	(15, 10, 8, 5)
Optimizer computed	—	(1.022e7, 1.042e5)	(1.022e7, 1.042e5)
Optimized value	—	17.70121235	17.70084996
Number of iterations	—	34	37
Backtracking Steps	—	94	143
CPU time	Several Days	3113s	2303s

Non-smooth: PMOR Framework is more efficient.

Floor Damper Design Problem: 2-norm Optimization

Table: Low Order Example

	Direct method	MOR Framework	PMOR Framework
Matrix size	280	25	(15, 10, 8, 5)
Optimizer computed	(1.102e7, 8.587e4)	(1.102e7, 8.587e4)	(1.102e7, 8.587e4)
Optimized value	6542.357196	6542.357196	6542.357196
Number of iterations	27	27	27
Backtracking Steps	35	30	29
CPU time	40s	7s	13s

Table: High Order Example

	Direct method	MOR Framework	PMOR Framework
Matrix size	29800	25	(15, 10, 8, 5)
Optimizer computed	—	(1.150e7, 9.189e4)	(1.150e7, 9.189e4)
Optimized value	—	710.738381	710.738381
Number of iterations	—	34	34
Backtracking Steps	—	36	36
CPU time	Several Days	1776s	2380s

Smooth: MOR Framework is more efficient.

Outline

Conclusion

- Design optimization of large scale dynamic systems is usually very expensive.
- Two-sided (P)MOR can compute both function values and gradients with moment matching properties, and thus can be used to reduce the computational cost of Quasi-Newton type optimization.
- MOR Framework and (P)MOR Framework are both effective in accelerating Quasi-Newton type optimization.
- For smooth objective functions, MOR Framework is more efficient.
- For non-smooth objective functions, PMOR Framework is more efficient.

Thank you!